Onderzoeksrapport Nr. 7906

THE GENERATION OF RANDOM ACTIVITY NETWORKS*

by

Willy S. HERROELEN**
Gerard CAESTECKER

March 1979

Wettelijk Depot : D/1979/2376/14

* Parts of this paper are based on Mr. Caestecker's MBA Thesis, K.U. Leuven, Belgium.

** Department of Applied Economics, K.U. Leuven, Belgium.

The authors appreciate the fruitful discussions with Prof. Dr. S. Elmaghraby on this topic.
This research was partially supported by Grant No.OT/V/19 with the Katholieke Universiteit Leuven.

ABSTRACT

Computational experiments conducted in the field of activity network planning techniques are usually based on the use of computer programs to generate a set of random activity networks. After discussing the basic characteristics to be possessed by a network generator in order to yield correct results, we describe two algorithms for generating a random topological structure of an activity network with a given number of nodes and arcs. These algorithms are incorporated in an overall network generator for the generation of a set of activity networks characterized by a suitable range of the number of nodes and arcs and corresponding topological structures.

1. INTRODUCTION

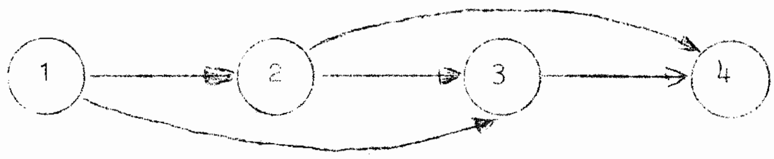
Many research efforts in the area of activity network planning techniques require the generation of random activity networks for the purpose of algorithm validation. A classical validation method implies the solution of a large number of representative problems where the required computing time and computer memory are used to measure the efficiency of the algorithms involved. The set of representative problems is usually obtained using a network generator to generate the topological structure of the networks, i.e., the underlying graph, together with a set of random values representing suitable outcomes for the functions associated with the graph (activity durations, resource requirements, etc.). Consequently, the many researchers in the area of activity networks ([2-5], [12], [13], [16-18]) and other complex problems involving network structures ([1], [6], [9-11], [14], [15]) have spent considerable effort in writing their own network generator. A critical examination of many well-known network generators forced Gendreau [9] towards the conclusion that a clear definition of the notion of a random network seems to be missing, resulting in the fact that many generators use construction rules which are centered around a rather loose statistical objective. The same author also makes a distinction between generators of strongly-random and weakly-random networks. Generators of weakly-random networks only use random number generators to yield the associated functions for networks with a given topological structure. A generator of strongly-random networks will generate the topological structure as well.

In generating activity networks the main difficulties do not seem to be caused by the random generation of the associated network functions (activity durations, resource requirements, etc.), but are mainly centered around the generation of random topological structures (see. also [8]). The purpose of this paper is to present a computer algorithm for generating a set of strongly-random activity networks, where each network is characterized by a certain number of nodes and arcs and a random topological structure. After an explicit problem statement in the next section, § 3 will focus on a computer algorithm for generating a single activity network with a given number of nodes and arcs. The overall procedure for generating a set of networks will then be discussed in § 4. The three Appendices concluding the paper give the corresponding subroutines for the algorithms of § 3, programmed in FORTRAN IV for an IBM 370/158 computer.

2. DETAILED PROBLEM STATEMENT

The topological structure of an activity network (further abbreviated as AN) consists of an acyclic directed graph $G = (N, A)$, where N denotes the number of nodes and A denotes the number of arcs. In the sequel, we assume that the AN has one source node and one sink node and that the nodes are numbered such that an arc always leads from a smaller number to a larger one. We further assume the activity-on-the-arc representation, i.e., the nodes represent the network events and the arcs denote the network activities.

An immediate consequence of the above mentioned numbering scheme is that the adjacency matrix representation of $G(N, A)$ is always upper triangular with zero diagonal. A typical AN and corresponding adjacency matrix are given in Figure 1 for $N = 4$ and $A = 5$.



(a) AN with N = 4 and A = 5

$$M = (m_{ij}) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 0 & 1 & 1 & 0 \\ & 0 & 1 & 1 \\ & & 0 & 1 \\ & & & 0 \end{array}\right] \end{array}$$

(b) Adjacency matrix

Figure 1 : Typical AN and adjacency matrix.

Another consequence of the above definitions and assumptions is that for a given N and A, where $(N-1) < A < \frac{N(N-1)}{2}$, several feasible G(N,A) may be generated. Figure 2 lists the other three alternative topological structures for N = 4 and A = 5. Consequently, the generation of a strongly-random G(N, A) for fixed N and A implies that the resultant topological network structures should have equal probabilities of occurrence. A procedure that satisfies this requirement is described in the next section.

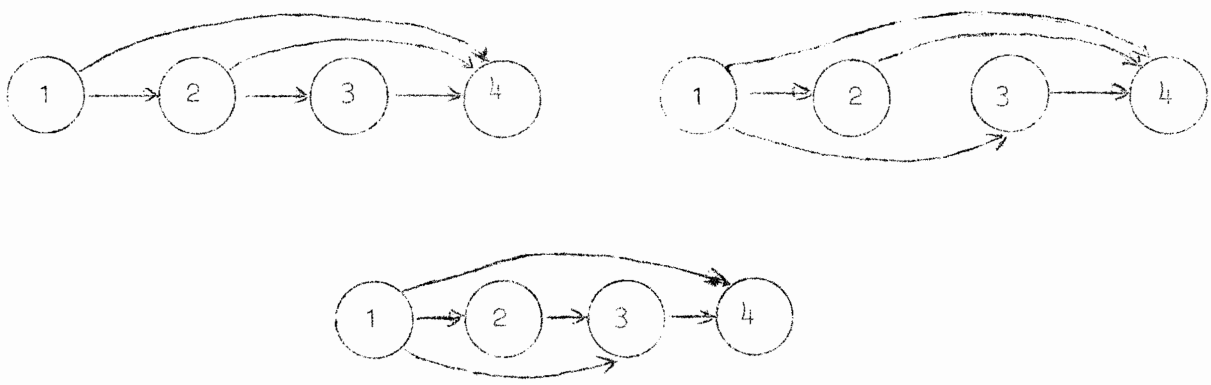


Figure 2 : Alternative topological structures for $N = 4$ and $A = 5$.

3. GENERATING ANs WITH FIXED N AND A.

The explicit generation of all possible topological structures for a G(N, A) proves to be an onerous task, not only because for specific combinations of N and A the number of feasible topological structures may be excessive, but also because the count of the number itself already turns out to be very cumbersome.

3.1. Counting the number of network structures.

Consider the adjacency matrix $M = (m_{ij})$ corresponding to a completely connected network G(N, A). For each row i of M, we define $n_i \triangleq \sum_j m_{ij}$. In order to reduce the completely connected N-node network to a network with only $A < \frac{N(N-1)}{2}$ arcs, we have to delete $D = \frac{N(N-1)}{2} - A$ arcs or, equivalently, D ones in the adjacency matrix, subject to the condition that every node in the final network except the sink emits at least one arc, and every node except the source is reached by at least one arc.

For a network with N nodes and A arcs, the total number of alternative topological network structures, K, can then be computed by summing the arc deletion combinations ; i.e.

$$K = \sum_{i=1}^{N-2} \sum_{k_i=0}^{n_i-1} \binom{n_i}{k_i} , \qquad (3.1)$$

subject to the conditions that $\sum_{i=1}^{N-2} k_i = D$, no row or column is left void, and arcs (1,2) and (N-1, N) are never deleted.

As an example, Figure 3 illustrates the computation of the number of topological structures for an AN with N = 5 and A = 5. Figure 3(a) represents the adjacency matrix corresponding to the completely connected network. The circled ones correspond to the requirement that arcs (1,2) and (4,5) may never be deleted. Figure 3(b) lists the several feasible k_i-values corresponding to the deletion requirement for D = 5 arcs. Applying Eq. (3.1) yields 11 different topological structures, for which the corresponding adjacency matrices are listed in Figure 3(c).

3.2. The deletion method.

Consider again the adjacency matrix $M = (m_{ij})$ corresponding to the completely connected N-node AN and define

$$n_i \triangleq \sum_j m_{ij} = N-i \ , \quad \text{for each row } i,$$

and

$$n_j \triangleq \sum_i m_{ij} = j-1, \text{ for each column } j.$$

	1	2	3	4	5
1	–	1	1	1	1
2		–	1	1	1
3			–	1	1
4				–	1
5					–

(a) Adjacency matrix

k_1 : 3 ; 3 ; 2
k_2 : 2 ; 1 ; 2
k_3 : 0 ; 1 ; 1

(b) k_i-values

	1	2	3	4	5
1	–	1	0	0	0
2		–	1	0	0
3			–	1	1
4				–	1
5					–

	1	2	3	4	5
1	–	1	0	0	0
2		–	1	0	1
3			–	1	0
4				–	1
5					–

	1	2	3	4	5
1	–	1	0	0	0
2		–	1	1	0
3			–	1	0
4				–	1
5					–

	1	2	3	4	5
1	–	1	0	0	0
2		–	1	1	0
3			–	0	1
4				–	1
5					–

	1	2	3	4	5
1	–	1	1	0	0
2		–	1	0	0
3			–	1	0
4				–	1
5					–

	1	2	3	4	5
1	–	1	1	0	0
2		–	0	1	0
3			–	1	0
4				–	1
5					–

	1	2	3	4	5
1	–	1	1	0	0
2		–	0	1	0
3			–	0	1
4				–	1
5					–

	1	2	3	4	5
1	–	1	1	0	0
2		–	0	0	1
3			–	1	0
4				–	1
5					–

	1	2	3	4	5
1	–	1	0	1	0
2		–	1	0	0
3			–	1	0
4				–	1
5					–

	1	2	3	4	5
1	–	1	0	1	0
2		–	1	0	0
3			–	0	1
4				–	1
5					–

	1	2	3	4	5
1	–	1	0	0	1
2		–	1	0	0
3			–	1	0
4				–	1
5					–

(c) Adjacency matrices for the possible network structures

Figure 3 : Alternative topological structures for N = 5, A = 5.

In other words, for a completely connected AN, n_i represents the total number of arcs leaving node i and n_j represents the total number of arcs entering node j.

A possible procedure for generating the topological structure of an AN with N nodes and A arcs involves the random <u>deletion</u> of $D = N(N-1)/2 - A$ ones in the adjacency matrix such that

$$\begin{aligned} n_i &\geq 1 \quad , \quad i = 1,2, \ldots, N-1 \\ n_i &= 0 \quad , \quad i = N \end{aligned} \tag{3.2}$$

and

$$\begin{aligned} n_j &= 0 \quad , \quad j = 1 \\ n_j &\geq 1 \quad , \quad j = 2,3, \ldots, N \end{aligned} \tag{3.3}$$

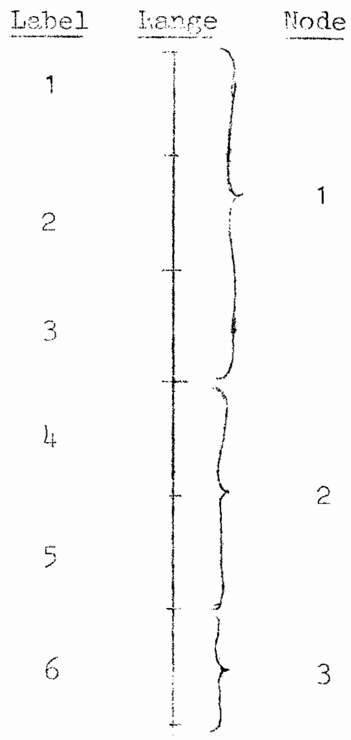
For any network G(N, A), the above conditions simply state that at least one arc must leave every node except the last, and at least one arc should enter every node except the first.

Due to the fact that the network generator should generate ANs with equal ex ante probabilities for the different feasible topological structures, all existing 1's in the adjacency matrix for the completely connected network should receive <u>equal deletion probabilities</u>, subject to the constraints (3.2) and (3.3).

This can be achieved by numbering all the 1's in the adjacency matrix for the completely connected AN from left to right and consecutively in the rows, as illustrated in Figure 4(a) for a 4-node network. The corresponding numbers (labels) are then assigned to equal intervals in the range of a uniformly distributed variable as illustrated in Figure 4(b). Drawing a random number will now yield an interval which in turn identifies the label of a corresponding arc.

	1	2	3	4
1		$^{1}(1)$	$^{1}(2)$	$^{1}(3)$
2			$^{1}(4)$	$^{1}(5)$
3				$^{1}(6)$
4				

(a)



(b)

Figure 4 : Label assignment.

In order to select a particular arc (i^*, j^*) to be deleted we proceed as follows. The sum of the label intervals corresponding to each node $\underline{i}$ equals the product of $(N - i)$ times the length of a label interval. For example in Figure 4, the intervals corresponding to node $i = 2$ sum up to $(4-2)\ (\frac{1}{6}) = 1/3$. It can also be seen from Figure 4 that $i = 2$ is preceded by 3 intervals of length $1/6$. In general the first label interval corresponding to a node i^* is preceded by at least

$$\sum_{0<i<i^*} (N-i) = (i^*-1)N - \frac{i^*(i^*-1)}{2} \qquad (3.4)$$

label intervals.

In order to generate an i^*, let $Y \sim U(0,1)$ and let

$$X = Y \frac{N(N-1)}{2} \tag{3.5}$$

where $\frac{N(N-1)}{2}$ denotes the total number of labels.

Now (3.4) and (3.5) imply that

$$X \geq (i^*-1)N - \frac{i^*(i^*-1)}{2}$$

or with $0 \leq \alpha < N-i^*$

$$\frac{i^{*2}}{2} - (N + \frac{1}{2})i^* + (N+X-\alpha) = 0,$$

which yields

$$i^* = (N + \frac{1}{2}) \pm \sqrt{(N + \frac{1}{2})^2 - 2(N+X-\alpha)} \tag{3.6}$$

Since $i^* \leq N - 1$, and $\alpha \geq 0$, Eq. (3.6) reduces to

$$i^* \leq (N + \frac{1}{2}) - \sqrt{(N + \frac{1}{2})^2 - 2N - 2X}$$

or

$$i^* \leq (N + \frac{1}{2}) - \sqrt{(N - \frac{1}{2})^2 - 2X}$$

Substituting Eq. (3.5) yields

$$i^* \leq (N + \frac{1}{2}) - \sqrt{N(N-1)(1-Y) + \frac{1}{4}}$$

Since $Y \sim U(0,1)$, $(1-Y) \sim U(0,1)$ which yields

$$i^* \leqslant N + \frac{1}{2} - \sqrt{N(N-1)Y + \frac{1}{4}} \qquad (3.7)$$

Since the last label interval corresponding to node i^* is followed by a number of label intervals at most equal to $i^*N - i^*(i^* + 1)/2$, similar arguments lead to

$$X \leqslant i^*N - \frac{i^*(i^*+1)}{2}$$

yielding

$$i^* \geqslant (N - \frac{1}{2}) - \sqrt{N(N-1)Y + \frac{1}{4}}$$

Hence

$$N - \frac{1}{2} - \sqrt{N(N-1)Y + \frac{1}{4}} \leqslant i^* \leqslant N + \frac{1}{2} - \sqrt{N(N-1)Y + \frac{1}{4}}$$

or

$$i^* = \left\lfloor N + \frac{1}{2} - \sqrt{N(N-1)Y + \frac{1}{4}} \right\rfloor , \qquad (3.9)$$

where $\lfloor a \rfloor$ denotes the greatest integer smaller than or equal to $\underline{a}$.

Given this value for i^*, we draw a new random observation of $Y \sim U(0,1)$ and rescale into $X \sim U(i^*+1,N+1)$ by setting

$$X = Y(N-i^*) + i^* + 1,$$

which in turn yields

$$j^* = \lfloor i^* + 1 + Y(N-i^*) \rfloor \tag{3.10}$$

The corresponding arc (i^*,j^*) can now be deleted from the network provided that conditions (3.2) and (3.3) are satisfied.

The deletion procedure for generating an activity network with N nodes and A arcs can now be stated as follows :

Step 0 (Initialization) : Read the number of nodes, N, and the number of arcs, A, and set up the adjacency matrix M corresponding to the completely connected network. Compute $D_{max} = N(N-1)/2$ and $D = D_{max} - A$. For each row i and column j of matrix M set $n_i = N-i$ and $n_j = j-1$, respectively.

Step 1 : Set t = 0 (the count of the deleted arcs).

Step 2 : Compute

$$i = \left\lfloor N + 1/2 - \sqrt{N(N-1)Y + 1/4} \right\rfloor ;$$

and

$$j = \lfloor i + 1 + X(N-i) \rfloor ,$$

where $X \sim U(0,1)$ and $Y \sim U(0,1)$. Set $i^* = i$, $j^* = j$ and $t = t+1$.

Step 3 : If the pair (i^*,j^*) has already been selected and/or if the conditions (3.2) and (3.3) are violated, reset $t = t-1$ and go to step 2. Otherwise, go to Step 4.

Step 4 : Update the adjacency matrix.

Step 5 : If $t = D$, Stop. Otherwise, return to Step 2.

This procedure has been programmed in FORTRAN IV for the IBM 370/158 computer. The corresponding subroutines are given in Appendix A.

3.3. The addition method.

The deletion method will delete a total of $\frac{N(N-1)}{2} - A$ arcs. For certain values of N and A this may be a very time consuming process. In order to generate a network with $N = 4$ and $A = 5$ for example, the deletion method will have to delete 1 arc ; however, if $N = 100$ and $A = 150$, 4800 out of the total $\frac{100(100-1)}{2} = 4950$ arcs need to be deleted. Under certain conditions, considerable time savings may be obtained by using a procedure which proceeds in the opposite direction; i.e., which starts from the adjacency matrix filled with zeros and adds the required number of ones. As a consequence of the node labeling procedure adopted, there should always be an arc connecting nodes 1 and 2 and an arc connecting nodes N-1 and N. Consequently such an addition procedure will have to generate a total of A-2 arcs. In view of all this, a good heuristic strategy would be to use the deletion method if $A \geqslant \frac{N(N-1)}{4}$ and to use the addition method if $A \leqslant \frac{N(N-1)}{4}$.

Consider again the problem of generating the topological structure of an AN with $N = 4$ and $A = 5$. Figure 5(a) represents the initial adjacency matrix, M, with $m_{12} = 1$ and $m_{34} = 1$ according to the requirement that there should be at least one arc leaving node 1 and one arc entering node 4. The corresponding network is given in Figure 5(b). The addition method will have to generate three additional arcs. However node 2 is not yet an emitting node and node 3 is not yet a receiving node.

$$M = (m_{ij}) = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} - & 1 & 0 & 0 \\ & - & 0 & 0 \\ & & - & 1 \\ & & & - \end{bmatrix} \end{array}$$

(a) Initial adjacency matrix

$$\textcircled{1} \longrightarrow \textcircled{2} \qquad \textcircled{3} \longrightarrow \textcircled{4}$$

(b) Initial topological structure.

Figure 5 : Adjacency matrix and corresponding topological structure.

This means that of the three additional arcs to be generated, only one may be inserted arbitrarily since in the final network at least one arc must leave node 2 and at least one arc must enter node 3. In general the initial network will be characterized by $m = N-3$ non-receiving nodes and $n = N-3$ non-emitting nodes. This means that $f = A-2-m-n$ arcs may be generated and inserted in a purely random fashion.

Consequently, the addition method will start from the initial network and adjacency matrix (all $m_{ij} = 0$ except $m_{12} = 1$ and $m_{N-1,N} = 1$) and may use Eqs. (3.9) and (3.10) to generate an arc as long as $f = A-e-m-n > 0$ where initially $e = 2$ (the count of the generated arcs), $m = N-3$ and $n = N-3$. Each time an arc is generated in this manner (and checked for double selection), the adjacency matrix is updated and $e = e+1$. If the generated arc reduces the number of non-receiving nodes we set $m = m-1$; if the number of non-emitting nodes is reduced we set

$n = n-1$. This process continues until either $A = e$, in which case the topological structure is completely defined and the procedure stops, or $f \leqslant 0$ and $e < A$, in which case we check whether $m = 0$.

If $m \neq 0$, this indicates that there is at least one non-receiving node : we locate that column, j^*, on the adjacency matrix that is completely filled with zeros (if ties develop, take the highest column index) and select a corresponding i^* between 1 and j^*-1 using equal probabilities $1/(j^*-1)$; i.e.,

$$i^* = \lfloor 1 + (j^*-1)Y \rfloor \quad , \quad \text{where } Y \sim U(0,1).$$

The addition of the corresponding arc (i^*, j^*) to the network requires updating of the adjacency matrix and the values of e, m and n.

If $m = 0$, we check whether there are still non-emitting nodes. If $n \neq 0$, we locate any zero row, i^*, in the adjacency matrix and generate a corresponding j^* between i^*+1 and N, using equal probabilities $1/(N-i^*)$; i.e.,

$$j^* = \lfloor i^* + 1 + (N-i^*)Y \rfloor \quad , \quad \text{where } Y \sim U(0,1).$$

The addition procedure for generating the topological structure of an AN with N nodes and A arcs can then be stated as follows :

Step 0 (Initialization) : Read the number of nodes, N, and the number of arcs, A, and set up the adjacency matrix, M, with $m_{ij} = 0$ except for m_{12} and $m_{N-1,N}$ which equal one. Set $e = 2$, and $m = n = N-3$. Compute $f = A - e - m - n$.

<u>Step 1</u> : If $e < A$, go to step 2; otherwise, <u>Stop</u>

<u>Step 2</u> : If $f \leqslant 0$, go to step 5; otherwise, go to step 3.

<u>Step 3</u> : Generate an arc (i^*, j^*) using Eqs. (3.9) and (3.10). Check for double selection.

<u>Step 4</u> : Set $e = e + 1$ and $m_{i^* j^*} = 1$. Update m and n and go to step 1.

<u>Step 5</u> : If $m = 0$, go to step 7; otherwise, go to step 6.

<u>Step 6</u> : Locate that column, j^*, on the adjacency matrix that is completely filled with zeros (if ties develop take the highest column index). Generate the arc (i^*, j^*), where

$$i^* = \lfloor 1 + (j^* - 1)Y \rfloor \qquad , Y \sim U(0,1)$$

Go to step 4.

<u>Step 7</u> : If $n = 0$, go to step [illegible]. Otherwise, locate an empty row, i^*, on the adjacency matrix and generate the arc (i^*, j^*), where

$$j^* = \lfloor i^* + 1 + (N - i^*)Y \rfloor \qquad , Y \sim U(0,1).$$

Go to step 4.

The corresponding FORTRAN IV subroutines are given in Appendix B. Appendix C gives the calling program for either the deletion method or the addition method.

4. GENERATING A SET OF STRONGLY RANDOM ANs

In the previous section two procedures were described for generating a feasible topological structure of an AN with a given number of nodes and arcs. However, as was mentioned in the introduction, many theoretical and practical situations require the use of a network generator for generating the topological structure of a set of ANs characterized by a representative range of the number of nodes, N, the number of arcs, A, and the topological structure (see [8]). More specifically, the generation of a set of ANs often implies the random selection of a set of (N,A) pairs, where for each pair several topological structures may be generated.

It was already argued above that for given values of the number of nodes, N, the number of arcs, A, is limited by $(N-1) \leq A \leq \frac{N(N-1)}{2}$. Figure 6 represents a matrix, obtained by applying the algorithm of Section 3.1, which lists the number of feasible topological structures for several (N, A) pairs. Since it would be too time consuming to generate all feasible topological structures for a given value of N, and even for a given (N, A) pair, a possible outcome would consist of determining the probability distribution of A given N, where for each value of N, a corresponding A-value can be obtained by drawing samples from the corresponding distribution.

It can be seen from Figure 6 that for $N \leq 3$, the p.d.f. has to assign equal probabilities to all feasible A-values. For $N = 4$ this equal probability assumption is no longer valid, but the p.d.f. of A given N is symmetric. For $N > 4$, however, the p.d.f. is no longer symmetric but shows a skewness to the right, where this skewness seems to increase with increasing N. Since obtaining an exact fit for this skew

N \ A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	1														
3		1	1												
4			1	4	4	1									
5				1	11	33	42	26	8	1					
6					1	26	168	508	881	845	584	263	76	13	1

Figure 6 : The number of feasible topological structures for several values of N and A.

distribution announces itself as a rather cumbersome task (the computational requirements of the counting procedure of Section 3.1 may become onerous for large values of N), we opt for the following heuristic procedure.

Figure 7 plots the range of the number of arcs, A, for increasing values of the number of nodes, N. The dotted curve represents the mean of the range on the number of arcs computed as

$$\left[\frac{N(N-1)}{2} + (N-1) \right] / 2 = \frac{1}{4} N^2 + \frac{1}{4} N - \frac{1}{2}.$$

Since the results of Figure 6 indicate that the observed mean of A lies below this theoretical mean, we have to adjust the latter. Therefore we set $\ell_A = N-1$ and $u_A = \frac{N(N-1)}{2}$ and compute the adjusted mean, μ_A as follows :

$$\mu_A = \ell_A + \frac{1}{2} (u_A - \ell_A) - \frac{3}{50} \left[(N-3)^{1/4} - 1 \right] (u_A - \ell_A),$$

which yields

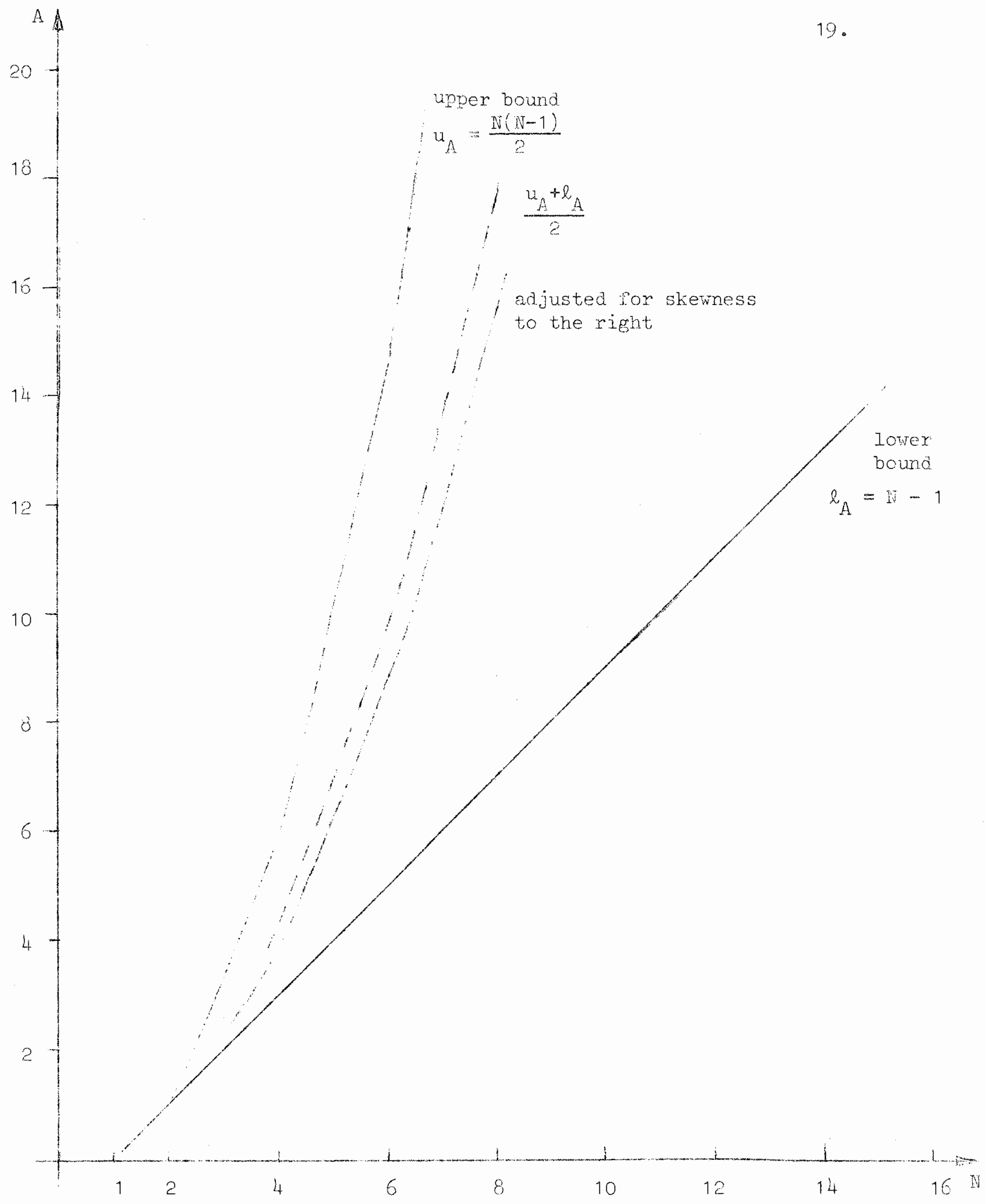


Figure 7 : Range of A for increasing values of N.

$$\mu_A = \ell_A + 2(u_A - \ell_A)\left[.28 - .03(N-3)^{1/4}\right]. \tag{4.1}$$

Setting

$$\sigma_A = \frac{\mu_A - N}{2}, \tag{4.2}$$

and given a value of N, a corresponding A-value is obtained by drawing a sample from the normal distribution with adjusted mean, μ_A, and adjusted standard deviation, σ_A, as given by Eqs. (4.1) and (4.2) respectively.

Given the so-obtained (N, A) pair, the algorithms of section 3 may be used to generate a strongly random topological structure. In [8] the above mentioned principles were used to generate a set of ANs to measure the so-called network complexity under the objective of computing the critical path. The N-values were generated by drawing samples from the modified exponential distribution (with $N \geq 3$ in order to obtain non-trivial networks), where

$$f(N) = 3 + \frac{1}{\lambda} e^{-\frac{1}{\lambda} N} \tag{4.3}$$

with $\lambda = \mu_N = \sigma_N = 50$. For each N-value, Eqs. (4.1) and (4.2) yielded the corresponding A-value(s), where several topological structures were generated for each (N, A) pair using the algorithms described in Section 3. For further details, we refer to [8].

5. CONCLUSIONS

Several practical experiments in the field of activity networks require the generation of a set of strongly-random networks, where each network is characterized by a certain number of nodes and arcs and a random topological structure. After discussing the necessary characteristics of a network generator, we have presented two algorithms - the addition method and the deletion method - for generating a random topological structure for a given value of the number of nodes and arcs. These algorithms may be integrated into a random network generator to generate a set of activity networks characterized by a representative range of the number of nodes and arcs and the corresponding topological structures.

Together with the classical principles for generating the associated network functions (see [2-5], [12], [13], [16-18]), this network generator may proof to fill a need in many computational experiments such as the measurement of network complexity (see [8]) and the validation of heuristic and analytical solution procedures for solving many combinatorial problems in the field of activity networks.

REFERENCES

[1] BENNINGTON, G.E., An efficient minimal cost flow algorithm, Management Science, 19, 1973, 1021-1051.

[2] COOPER, D.F., Heuristics for scheduling resource-constrained projects : An experimental investigation, Management Science, 22, 1976, 1186-1194.

[3] DAVIES, E.M., An experimental investigation of resource allocation in multiactivity projects, Operational Research Quarterly, 24 1974, 587-591.

[4] DAVIS, E.W., Project summary measures and constrained resource scheduling, AIIE Transactions, 7, 1975, 132-142.

[5] DAVIS, E.W., HEIDORN, G.E., An algorithm for optimal project scheduling under multiple resource constraints, Management Science, 17, 1971, B-803-816.

[6] DIAL, R., GLOVER, F., KARNEY, D., KLINGMAN, D., A computational analysis of alternative algorithms and labeling techniques for finding shortest path trees, Working Paper CCS 291, Centre for Cybernetic Studies, University of Texas, April 1977.

[7] DUSSAULT, J.P., Un générateur de réseaux aléatoires, Publication no 262, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Août 1977.

[8] ELMAGHRABY, S.E., HERROELEN, W.S., On the measurement of complexity in activity networks, OR Report No. 121, NC State University, Raleigh, September 1978.

[9] GENDREAU, M., Etude sur la génération des réseaux aléatoires, Document de travail no. 98, Département d'Informatique, Université de Montréal, Octobre 1977.

[10] GILSINN, J., WITZGALL, C., A performance comparison of labeling algorithms for calculating shortest path trees, NBS Technical Note 777, National Bureau of Standards, Washington, 1973.

[11] GOLDEN, B., Shortest-path algorithms : A comparison, Operations Research, 24, 1976, 1164-1168.

[12] GORDON, J.H., Project network categorization in an evaluation of heuristic resource allocation techniques, unpublished Ph.D Thesis, University of Birmingham, England, 1976.

[13] JOHNSON, T.J.R., An algorithm for the resource-constrained project scheduling problem, unpublished Ph.D.Thesis, M.I.T., 1967.

[14] KLINGMAN, D., NAPIER, A., STUTZ, J., NETGEN : A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems, <u>Management Science</u>, 20, 1974, 814-822.

[15] LOS, M., Simultaneous optimization of land use and transportation in new town design, unpublished Ph.D Thesis, Univ. of Pennsylvania, 1975.

[16] PASCOE, T.L., Allocation of resources-CPM, <u>Revue Française de Recherche Opérationnelle</u>, 38, 1966, 21-28.

[17] PATTERSON, J.H., Project scheduling : The effects of problem structure on heuristic performance, <u>Nav. Res. Log. Quart.</u>, 23, 1976, 95-123.

[18] THESEN, A., Measures of the restrictiveness of project networks, <u>Networks</u>, 7, 1977, 193-208.

APPENDIX A

FORTRAN SUBROUTINES FOR THE DELETION METHOD

```
      SUBRØUTINE GENAND(N,A,ADJ,IX)
C
C  THIS SUBRØUTINE GENERATES THE ADJACENCY
C  MATRIX ØF AN ACTIVITY NETWØRK BY THE DELETIØN
C  METHØD
C
C  N=NUMBER ØF NØDES (INT)
C  A=NUMBER ØF ARCS (INT*2)
C  ADJ(.,.)=ADJACENCY MATRIX (INT*2)
C  IX=RANDØM INTEGER
C
      INTEGER*2 A,ADJ(500,500),NRI(500),NRJ(500)
C
C  THIS DIMENSIØN STATEMENT ALLØWS FØR THE GENERATIØN
C  ØF ANS WITH A MAX.ØF N=500 NØDES AND
C  A=124750 ARCS
C
C  CØNSTRUCTIØN ØF CØMPLETELY CØNNECTED ADJACENCY
C  MATRIX
C
      NMIN1=N-1
      DØ 2 I=1,NMIN1
      NRI(I)=N-I
      NRJ(I)=I-1
      DØ 1 J=1,I
    1 ADJ(I,J)=0
      ADJ(N,I)=0
      IPLUS1=I+1
      DØ 2 J=IPLUS1,N
    2 ADJ(I,J)=1
      ADJ(N,N)=0
      NRJ(N)=N-1
C
C  CØMPUTE NR.ØF ARCS TØ BE DELETED
C
      M=N*NMIN1
      NR=0.5*M-A
      DØ 4 L=1,NR
```

```
C   CALL SUBRØUTINE TØ GENERATE I AND J
C
     3 CALL INTSEL(IX,I,J,N,M)
C
C   CHECK FØR DØUBLE SELECTIØN
       IF(ADJ(I,J).EQ.0) GØ TØ 3
C
C   CHECK FØR FEASIBILITY CØNDITIØNS
       IF (NRI(I).EQ.1) GØ TØ 3
       IF (NRJ(J).EQ.1) GØ TØ 3
       ADJ(I,J)=0
       NRI(I)=NRI(I)-1
     4 NRJ(J)=NRJ(J)-1
       RETURN
       END

       SUBRØUTINE INTSEL(IX,I,J,N,M)
C
C   THIS SUBRØUTINE SELECTS AN ARC (I,J)
C
       CALL RANDØM(IX,Y)
       I=N+0.5-SQRT(M*Y+0.25)
       CALL RANDØM(IX,Y)
       J=I+1+(N-I)*Y
       RETURN
       END
```

APPENDIX B

FORTRAN SUBROUTINES FOR THE ADDITION METHOD

```
      SUBRØUTINE GENANA(N,A,ADJ,IX)
C
C  THIS SUBRØUTINE GENERATES THE ADJACENCY
C  MATRIX ØF AN ACTIVITY NETWØRK BY THE
C  ADDITIØN METHØD
C  A=NUMBER ØF ARCS (INT*2)
C  N=NUMBER ØF NØDES (INT)
C  ADJ(.,.)=ADJACENCY MATRIX (INT*2)
C  IX=RANDØM INTEGER
C
      INTEGER*2 A,ADJ(500,500),NØDES(500)
C
C  THIS DIMENSIØN STATEMENT ALLØWS FØR THE
C  GENERATIØN ØF ANS WITH A MAX. ØF N=500
C  NØDES AND A=124750 ARCS
C
C  CØNSTRUCTIØN ØF INITIAL ADJACENCY MATRIX
C
      DØ 1 I=1,N
      NØDES(I)=0
      DØ 1 J=1,N
    1 ADJ(I,J)=0
      NØDES(1)=2
      NØDES(2)=3
      NØDES(N-1)=1
      NØDES(N)=2
      ADJ(1,2)=1
      ADJ(N-1,N)=1
C  INITIALIZE CØUNT ØF GENERATED ARCS
      L=2
C  INITIALIZE NR.ØF NØN-EMITTING AND NØN-
C  RECEIVING NØDES
      M1=N-3
      M2=M1
C
      M=N*(N-1)
      K=1
C  CHECK NR.ØF ARCS TØ BE GENERATED IN A
C  PURELY RANDØM FASHIØN
C
    2 IF(A-L-M1-M2.LE.0) GØ TØ 3
```

```
C
C   GENERATE ARC AND CHECK FØR DØUBLE SELECTIØN
C
      8 CALL INTSEL(IX,I,J,N,M)
        IF (ADJ(I,J).EQ.1) GØ TØ 8
C
C   CALL SUBRØUTINE INCTAL FØR UPDATING PURPØSES
        CALL INCTAL(ADJ,NØDES,L,I,J,M1,M2)
        GØ TØ 2
C
C   CHECK FØR NØN-RECEIVING NØDES AND GENERATE ARC
C
      3 IF (M1.EQ.0) GØ TØ 5
        M3=K+1
        DØ 4 K=M3,N
        J=N-K+1
        IF (NØDES(J).GT.1) GØ TØ 4
        CALL RANDØM(IX,Y)
        I=1+(J-1)*Y
        CALL INCTAL(ADJ,NØDES,L,I,J,M1,M2)
        GØ TØ 2
      4 CØNTINUE
C   CHECK FØR NØN-EMITTING NØDES AND GENERATE ARC
C
      5 IF (M2.EQ.0) GØ TØ 7
        DØ 6 I=2,N
        IF (NØDES(I).NE.3) GØ TØ 6
        CALL RANDØM(IX,Y)
        J=I+1+(N-I)*Y
        ADJ(I,J)=1
      6 CØNTINUE
      7 RETURN
        END

        SUBRØUTINE INCTAL(ADJ,NØDES,L,I,J,M1,M2)
C
C   THIS SUBRØUTINE WILL UPDATE CØUNTERS FØR
C   NR.ØF GENERATED ARCS,NR.ØF NØN-EMITTING
C   NØDES,NR.ØF NØN-RECEIVING NØDES AND ADJACENCY
C   MATRIX
C
        INTEGER*2 ADJ(500,500),NØDES(500,500)
C
```

```
  L=L+1
  ADJ(I,J)=1
  K=NØDES(I)
  GØ TØ (3,3,1),K
  NØDES(I)=1
  GØ TØ 2
1 NØDES(I)=2
2 M2=M2-1
3 K=NØDES(J)
  GØ TØ (4,6,6),K
  NØDES(J)=3
  GØ TØ 5
4 NØDES(J)=2
5 M1=M1-1
6 RETURN
  END
```

APPENDIX C

CALLING PROGRAM FOR ADDITION OR DELETION METHOD

```
      SUBRØUTINE GENAN(N,A,ADJ,IX)
C
      INTEGER*2 A,ADJ(500,500)
C
      IF(A-N*(N-1)*0.25) 1,1,2
    1 CALL GENANA(N,A,ADJ,IX)
      RETURN
      END
C
    2 CALL GENAND(N,A,ADJ,IX)
      RETURN
      END
```